Sinclair ZX SPECTRUM Userclub

Clubinfo Nr. 9 (114) vom 01. September 1993

Rolf Knorre - Postfach 20 10 38 - 42210 Wuppertal
Telefon 0202 / 55 66 89 - Telefax 0202 / 59 47 21
Stadtsparkasse Wuppertal (BLZ 33050000) Kto. 853382

Hallo Freunde,

auch diesen Monat besteht das Info wieder aus alten und neuen Beiträgen rund um den Spectrum. Das ich den Umfang des Infos gerne steigern würde, habe ich ja bereits mehrfach erwähnt. Meine Aufrufe haben bisher noch nicht gefruchtet. Leider bin ich mir nicht sicher, ob eine 8-wöchige Erscheinungsweise daran etwas ändern würde. Die bisherigen Reaktionen aus dem Club zeigen aber eine grundsätzliche Zustimmung. Ich warte noch weitere Stimmen ab. Viel Spaß mit dem neuen Info.

Rolf Knorre

P.S. Um Kosten zu sparen, erbitte ich Überweisungen der Clubbeiträge verstärkt auf meine Postscheckkonto Köln (BLZ 37010050) Konto Nr. 198628-502. Danke.

Hubert Kracher
Kirchdorfer Str. 5
83064 Großholzhausen

Harald Lack
Heidenauer Str. 5
83064 Raubling

* DOUBLE 'H' SOFTWARE *

Liebe Adventure-Freunde!!

Mit dem heutigen Beitrag haben wir dann bereits mehr als die Hälfte des Adventures "LORDS OF TIME" gelost und deshalb wollen wir uns auch nicht lange mit Einfuhrungen aufhalten, sondern sogleich an den letzten Teil anknupfen. Wir befinden uns jetzt in der Zeitzone 5, der Tudor Ara. Wie bereits bestens bekannt, haben wir die Zeitmaschine wieder in Richtung Norden verlassen und befinden uns "at the end of a road leading north". Weitere Locations in diesem Teil sind:

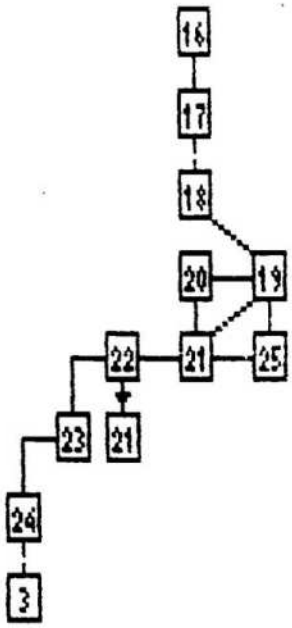
04) AT THE END OF A ROAD LEADING NORTH
05) AT A BEND IN THE ENCLOSED ROAD / MILESTONE
06) ON A ROAD ACROSS A SMALL VILLAGE
07) ON A COBBLED STREET
08) AT THE COBBLE SQUARE / ROTEN APPLE, WRETCH
09) IN FRONT OF THE IMPOSING CASTLE / THIRSTY MESSENGER
10) NORTH BANK OF THE MOAT
11) IN A COPSE A
12) AT THE EDGE OF THE COPSE
13) DEEP IN THE COPSE
14) IN THE COPSE B / FIRE BREATHING DRAGON
15) IN THE BAR / CASK OF ALE, ROTUND BARTENDER
16) IN AKLE DEEP SCUMMY WATER / DRAGON'S WING, FROG
17) AT THE CASTLE ENTRANCE
18) IN THE COURTYARD / GAUNTLET
19) KEEP ENTRANCE
20) IN THE ARMOURY / BATTERED SUIT
21) RUSH-STREWN GREAT HALL
22) ABOVE A NARROW FLIGHT OF STAIRS
23) DESCENDING STAIRS
24) IN THE DUNGEONS / COFFER
25) STORE ROOM

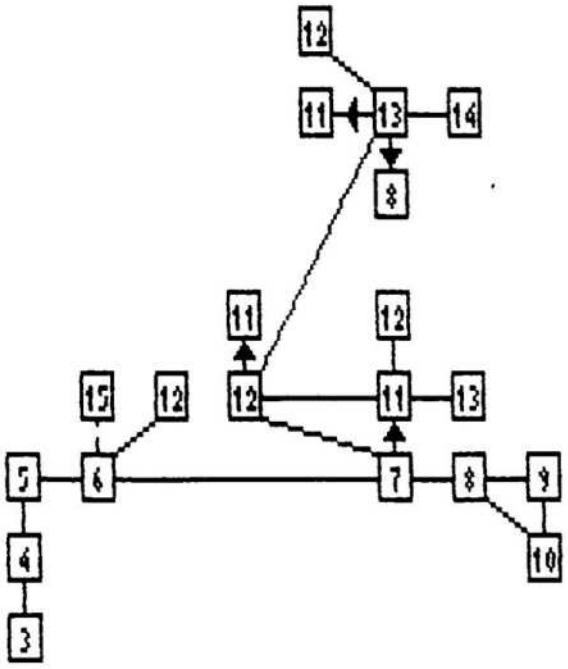
Wir machen folgendermaßen weiter:

N, examine milestone (darauf steht Hope Village, das ist die Hoffnung, die es in Zeitzone 9 aufzugeben gilt), take milestone, E, open door, in, take ale (der Schankkellner verlangt Geld dafür), give nugget, take ale, open door, out, E, E, E (wir treffen einen Kurier des Königs, der sehr durstig ist), give ale (dafür bekommen wir was zu essen), examine food (es ist eine ideale Drachennahrung), W, W, N, E, E, give food (der Drachen frißt das Futter und fragt ob wir auf seinem Rücken mitfliegen wollen), yes (der Drache fliegt mit uns über den Graben und landet am Südufer), take dragons wing, take frog, kiss frog (der Frosch verwandelt sich in einen hübschen Prinzen, der uns von jetzt an folgt), S, in, take gauntlet, SE, W, take armour, wear armour, S (wir sehen den Schwarzen Ritter), W (der Schwarze Ritter zieht sein Schwert), W (der Schwarze Ritter greift uns an, aber der Prinz springt dazwischen, tötet ihn und bringt seine Leiche weg), W, D, D, dig (wir sind in einer Grube), dig (wir graben eine juwelenbesetzte Kassette aus), take coffer, dig (wir sind wieder in der Zeitmaschine und trennen uns von folgenden Dingen:

drop coffer, drop armour, drop gauntlet, drop dragons wing, drop milestone. Score bringt uns nun hoffentlich den Stand 475 out of 1000. Ab geht es in Zeitzone 6, turn cog 6, swing pendulum, N.....

Soviel wieder einmal dazu. Fortsetzung folgt im nachsten Info.
Viel Spaß bis dahin beim Ausprobieren.





CORNERPACK(S) by Sherlo

Cornerpacks erfüllen einen ähnlichen Zweck wie die im Teil 18 erwähnten Borderpacks. Die Cornerpacks sind im Gegensatz zu den Borderfiles reine Screenfiles (**BILD 1 + 2**) und werden je nach Größe des Rahmen zusammengestellt. **START:** Als erstes wird ein **"CP"** in den Speicher von WM geladen. In diesem Info habe ich den **"CP 74"**

```
Graphics
*?4iu
*?4ru
*?4io
*?4ro
*?4o
*?4s1
*?4u
*?4sr
*?4in
*?4rn
*?4s1
*?4sr
```

ausgesucht. Genau wie bei den Borderpacks empfielt es sich alle Cornerpacks auszudrucken. Zusätzlich zum **"CP 74"** benötigen wir einen **Text**, den **typliner!** sowie einen **3er Font** (z.B. Title) im Speicher von WM. 1.) "G"(et) drücken - **typliner!** eingeben + **Enter**. Nun **"CP74A4"** eingeben und die Frage "**"Get Page"** mit **"y"**(es) beantworten. Jetzt müßt Ihr den **Namen Eures Textfiles** eingeben + **Enter**. Es erscheint der Typmodus-Screen - **"EDIT"** drücken um den 3er Font einzugeben. Wenn jetzt die Taste **"V"** sowie anschließend Taste **"P"** gedrückt wird setzt sich der

Drucker in Bewegung. 2)**INV-VIDEO** drücken, **CP74A5q** eingeben + **Enter** - **Get Page?** mit **"y"** beantworten - **EDIT** und den **3er Font eingeben** - Taste **"V"** und **"P" drücken.** 3.) Die beiden noch verbliebenen **>Fpge**-Files **CP-74A5h** und **CP74vc** werden wie unter Punkt 2 schon beschrieben ausgedruckt.

TYG! by Sherlo

Tyg! ist ein auf 23 (!) **Graphics** (**BILD 3**) erweiterter Typliner. Die Frage wozu man 23 Grafikspeicherplätze benötigt ist schnell beantwortet. Gerade wenn Ihr mit den Borderpacks arbeitet könnt Ihr auf diese Speicherplätze nicht verzichten. Der Rahmen für dieses Info hat allein schon 12 Speicherplätze (**BILD 1**) "geschluckt". Tyg! braucht etwas mehr Speicherplatz (370 BYTES) als der normale Typliner, aber 370 bytes lassen sich ohne Probleme verkraften. Die Typliner - Funktionen sind ansonsten identisch mit Tyg! **ACHTUNG:** Es gibt jedoch ein kleines Problem wenn vom Typliner! erstellte **>Fpge**-Files im Tyg! aktiviert werden bzw. auch umgekehrt! Deshalb solltet Ihr unbedingt darauf achten, daß Ihr

```
Graphics
*?4iu
*?4ru
*?4io
*?4ro
*?4o
*?4s1
*?4u
*?4sr
*?4in
*?4rn
*?4s1
*?4sr
u43
COLFILE2    sh
bild1
colfile5    sh
COLFILE2    sh
*?4in
COLFILE2    sh
bild3
COLFILE7    sh
COLFILE7    sh
```

Eure **>Fpge** - Files entsprechend gekennzeichnet habt. Somit kann es zu keiner Verwechslung bzw. unnötigen Ärger kommen. Euer Speccy stürzt Euch sonst total ab und somit natürlich auch der gesamte Speicherinhalt.

ENDE

Im nächsten Teil geht es um den dreiteiligen Font **"ANTHEA 3"**, aber bis dahin wünsche ich Euch noch viel Spaß mit DTP LG 22.

G. **Marten** **03.08.93**

Günther Marten Neue Straße 3 2900 Oldenburg Telefon 0441 - 17976

* DTP LEICHT GEMACHT 23 * DOPPEL- UND MEHRFACHFONTS TEIL 7 *

Der DTP-Font "Anthea 3" hat genau wie der Font Modern 3 viele Kombinations-Möglichkeiten. Die drei folgenden Beispiele zeigen nur einen kleinen Teil der Möglichkeiten.

Der dreiteilige DTP - Font "ANTHEA 3":
Dieser DTP - Font arbeitet nur mit Großbuchstaben. Sonstige Zeichen wie Punkt, Komma, Plus, Minus etc. wurden nicht berücksichtigt! Bild 1 (rechts) zeigt wie das Demo **"ANTHEA 3"** zustande gekommen ist. HINWEIS: SS bedeutet SYMBOL-SHIFT. Es folgt die Tastenbelegung:

t.Anthea: A = ANSTELLE EINES BUCHSTABEN • m = ZWISCHEN DEN BUCHSTABEN

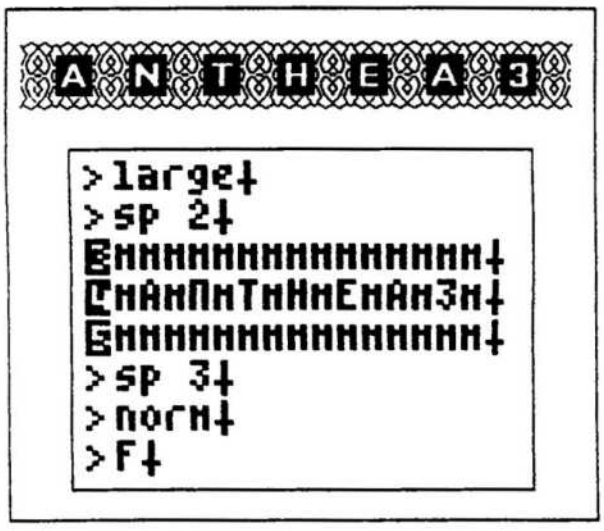
m.Anthea: Die Tasten A - Z sowie 0 - 9 sind mit der Spectrumtastatur identisch. Ä = SS + Y •• Ö = SS + O •• Ü = SS + U •• PUNKT = LEERKASTEN •• m = ZWISCHEN DEN BUCHSTABEN

b.Anthea: SIEHE t. Anthea

t.Anthea

m.Anthea A B C D E F G H I J K L O P Q 1 2 3 4 5 6 7 8 9 0 Ä Ö Ü

b.Anthea



Im nächsten Teil geht es mit dem **DOKUFONT!. CA! und DEL!** weiter, aber bis dahin wünsche ich noch viel Spaß mit DTP LG Teil 23. Günther Marten Oldenburg, den 01. 09. 93

Spectrum User Club Wuppertal

Wolfgang Bernut, Friedberger Straße 92c, 6360 Friedberg 2
Telefon: 06031/2963

Friedberg, den 28.12.1989

Hallo Spectrum-User!

In dieser Ausgabe soll nun der Schluß des PASCAL-Programms zur Aufnahme von Kennlinien erscheinen. Es würde mich freuen, wenn jemand einen Nutzen daraus ziehen könnte. Vielleicht sind auch Teile des Programms für ganz andere Problemstellungen zu gebrauchen. Im Stillen erhoffe ich mir auch Kontakt zu Mitgliedern, die ebenfalls mit HISOFT-PASCAL arbeiten.

Was unseren Spectrum-User-Club betrifft, wäre es schön, wenn er noch lange bestehen könnte. Ideen gibt es doch sicher noch genug?! Vielleicht macht der neue SAM einigen Mitgliedern Mut, den Spectrum noch nicht zum alten Eisen zu legen. Ich jedenfalls bin neugierig, meine Programme einmal mit einer besseren Graphik zu sehen, die der SAM ja wohl haben soll. Trotzdem kann man sagen, daß die Spectrum-Graphik für die meisten Fälle ausreicht. Ich finde, daß das Ergebnis einer Arbeit entscheidend ist, das man am Schluß in den Händen hält. Und da kann man mit dem Spectrum zu hervorragenden Ergebnissen kommen. Man denke nur an das DESK-TOP-PUBLISHING.

Ich finde auch, man sollte sich lieber einen guten Drucker kaufen und sich an der schönen Schrift erfreuen, als sein Geld für einen Ultra-Turbo-Computer auszugeben und an einem Drucker zu sparen. So mancher einfache Cassettenrecorder bringt mit guten Bändern mehr Klang als ein Super-Sound-Recorder mit billigen Wegwerfcassetten.

Auf, Spectrum-User, der Kampf gegen die Computer-Goliaths ist noch nicht verloren. Kauft Euch ruhig einen PC oder AT zum Spielen und Vorzeigen für Freunde und Nachbarn, aber mit dem Spectrum wird *gearbeitet*! Ich denke, daß wir uns verstehen.

Zum Schluß noch ein herzliches Danke an Rolf Knorre, der mit großem Einsatz den Club betreut.

Herzliche Grüße

```
. 20  PROCEDURE sammeln(x,y:INTEGER);
 930
3940  { Funktionswerte werden nach Steile geordnet addiert }
3950
3960  BEGIN
3970    summe[x]:=summe[x]+y;
3980    n[x]:=n[x]+1;
3990  END; (* sammein *)
4000
4010
4020  PROCEDURE werteeinlesen;
4030
4040  { steuert Spannung des programmierbaren  Netzgeraets und  misst  Spannung  am }
4050  { Vorwiderstand, woraus Spannung und  Stromstaerke  am  Messobjekt  berechnet }
4060  { werden; die Punkte P(U;I) werden ins Koordinatensystem gezeichnet }
4070
4080  CONST uaus           = 5.1;              (* muss vorher gemessen werden      *)
4090        uein           = 5.1;              (* muss vorher gemessen werden      *)
4100        vorwiderstand = 38;
4110  VAR   i,
4120        adu,
4130        zaehler,
4140        xkoord,ykoord : INTEGER;
4150        grenze        : REAL;
4160  BEGIN
4170    koordinatensystem;
4180    initpio;
4190    initsumme;
4200    grenze :=imax*vorwiderstand*255/uein;
4210    zaehler:=0;
4220    REPEAT
4230      FOR i:=1 TO 3 DO
4240        BEGIN
4250          OUT(159,CHR(zaehler));          (* PIO-Port A steuert D/A-Umsetzer  *)
4260          delay(5);
4270          OUT( 63,CHR(0));                (* Startimpuls fuer A/D-Umsetzer    *)
4280          delay(5);
4290          adu   :=ORD(INP(127));          (* lesen des A/D-Umsetzer-Registers *)
4300          xkoord:=ROUND((zaehler*umax*uaus-adu*uein)/255*100/uachse);
4310          ykoord:=ROUND(adu/255*uein/vorwiderstand*1000*iachse);
4320          IF (xkoord>=0) AND (xkoord<=255)
4330            THEN
4340              BEGIN
4350                IF steil
4360                  THEN
4370                    sammeln(ykoord,xkoord)
4380                  ELSE
4390                    sammeln(xkoord,ykoord);
4400                IF (ykoord>=0) AND (ykoord<=167)
4410                  THEN
4420                    plot(xkoord,ykoord);  (* entspr. Punkt P(U;I) Zeile 4060  *)
4430              END (* if then *)
4440            ELSE
4450              IF xkoord>255
4460                THEN
4470                  BEGIN
4480                    OUT(159,CHR(0));      (* Ausgangsspannung auf 0 V setzen  *)
4490                    zaehler:=255;
4500                  END; (* if then else if *)
4510          END; (* for *)
4520      zaehler:=zaehler+1
4530    UNTIL (zaehler=256) OR (adu>=grenze) OR (adu>254);
4540    OUT(159,CHR( 0));                     (* Ausgangsspannung auf 0 V setzen  *)
4550  END; (* werteeinlesen *)
4560
4570
```

```
4580  PROCEDURE initwert;
4590
4600  { loescht alle Punkte P(x;wert[x]) }
4610
4620  VAR x:INTEGER;
4630  BEGIN
4640    FOR x:=0 TO 255 DO
4650      wert[x]:=-1;
4660  END; (* initwert *)
4670
4680
4690  PROCEDURE berechnen;
4700
4710  { berechnet Mittel ueber gleiche x-Werte (oder gleiche y-Werte  falls  steil= }
4720  { TRUE ist; es wurden dann xkoord und ykoord in 'werteeinlesen' getauscht) }
4730
4740  VAR x:INTEGER;
4750  BEGIN
4760    initwert;
4770    FOR x:=0 TO 255 DO
4780      BEGIN
4790        IF n[x]>0
4800          THEN
4810            BEGIN
4820              wert[x]:=ROUND(summe[x]/n[x]);
4830              IF wert[x]<=167
4840                THEN
4850                  xmax:=x;
4860            END; (* if *)
4870      END; (* for *)
4880  END; (* berechnen *)
4890
4900
4910  PROCEDURE neuberechnen;
4920
4930  { berechnet neue Werte, falls in 'berechnen' ueber gleiche  y-Werte gemittelt }
4940
4950  VAR x,y:INTEGER;
4960  BEGIN
4970    initsumme;
4980    FOR y:=0 TO xmax DO
4990      IF wert[y]<>-1
5000        THEN
5010          BEGIN
5020            x:=wert[y];
5030            sammeln(x,y);
5040          END; (* if *)
5050    berechnen;
5060  END; (* neuberechnen *)
5070
5080
```

```
5090  FUNCTION dreipunkt(a,b,c,d,e:INTEGER):BOOLEAN;
5100
5110  { stellt fest, ob Luecke drei Punkte breit ist }
5120
5130  BEGIN
5140    dreipunkt:=(a<>-1) AND (b=-1) AND (c=-1) AND (d=-1) AND (e<>-1)
5150  END; (* dreipunkt *)
5160
5170
5180  FUNCTION zweipunkt(a,b,c,d:INTEGER):BOOLEAN;
5190
5200  BEGIN
5210    zweipunkt:=(a<>-1) AND (b=-1) AND (c=-1) AND (d<>-1)
5220  END; (* zweipunkt *)
5230
5240
5250  FUNCTION einpunkt(a,b,c:INTEGER):BOOLEAN;
5260
5270  BEGIN
5280    einpunkt:=(a<>-1) AND (b=-1) AND (c<>-1)
5290  END; (* einpunkt *)
5300
5310
5320  PROCEDURE lueckenweg;
5330
5340  { berechnet fehlende Punkte }
5350
5360  VAR x,dify: INTEGER;
5370  BEGIN
5380    FOR x:=0 TO xmax-4 DO
5390      IF dreipunkt(wert[x],wert[x+1],wert[x+2],wert[x+3],wert[x+4])
5400        THEN
5410          wert[x+2]:=ROUND((wert[x]+wert[x+4])/2);
5420    FOR x:=0 TO xmax-3 DO
5430      IF zweipunkt(wert[x],wert[x+1],wert[x+2],wert[x+3])
5440        THEN
5450          BEGIN
5460            dify:=wert[x+3]-wert[x];
5470            wert[x+1]:=ROUND(wert[x]+1*dify/3);
5480            wert[x+2]:=ROUND(wert[x]+2*dify/3);
5490          END; (* if *)
5500    FOR x:=0 TO xmax-2 DO
5510      IF einpunkt(wert[x],wert[x+1],wert[x+2])
5520        THEN
5530          wert[x+1]:=ROUND(wert[x]+(wert[x+2]-wert[x])/2);
5540  END; (* lueckenweg *)
5550
5560
```

```
5570  PROCEDURE pruefexmax;
5580
5590  { entfernt letzten Punkt, falls er einen zu niedrigen Funktionswert hat }
5600
5610  BEGIN
5620    IF wert[xmax]<wert[xmax-1]
5630      THEN
5640        BEGIN
5650          wert[xmax]:=-1;
5660          xmax:=xmax-1;
5670        END; (* if *)
5680  END; (* pruefexmax *)
5690
5700
5710  PROCEDURE glaetten(zumrunden:REAL);
5720
5730  { entfernt Zacken des Graphen }
5740
5750  VAR neuwert:ARRAY[1..254] OF INTEGER;
5760      x:INTEGER;
5770  BEGIN
5780    pruefexmax;
5790    FOR x:=1 TO xmax-1 DO
5800      neuwert[x]:=-1;
5810    FOR x:=0 TO xmax-2 DO
5820      neuwert[x+1]:=TRUNC(wert[x]+(wert[x+2]-wert[x])/2+zumrunden);
5830    FOR x:=1 TO xmax-1 DO
5840      IF neuwert[x]<>-1
5850        THEN
5860          wert[x]:=neuwert[x];
5870  END; (* glaetten *)
5880
5890
5900  PROCEDURE zeichnen;
5910
5920  { zeichnet alle Punkte P(x;wert[x]) von x=0 bis x=xmax }
5930
5940  VAR x: INTEGER;
5950  BEGIN
5960    koordinatensystem;
5970    FOR x:= 0 TO xmax DO
5980      plot(x,wert[x]);
5990  END; (* zeichnen *)
6000
6010
```

```
6020  PROCEDURE verbinden;
6030
6040  { fuellt bei groesserer Steigung noch vorhandene Luecken aus }
6050
6060  VAR x,dify:INTEGER;
6070  BEGIN
6080    FOR x:=0 TO xmax-1 DO
6090      BEGIN
6100        dify:=wert[x+1]-wert[x];
6110        IF dify>1
6120          THEN
6130            BEGIN
6140              plot(x,wert[x]);
6150              linie(1,dify);
6160            END; (* if *)
6170      END; (* for *)
6180  END; (* verbinden *)
6190
6200
6210  PROCEDURE andersverbinden;
6220
6230  { fuellt grosse Luecken aus, die nach 'neuberechnen' entstehen koennen }
6240
6250  VAR x:INTEGER;
6260  BEGIN
6270    wert[0]:=0;
6280    koordinatensystem;
6290    FOR x:=0 TO xmax DO
6300      IF wert[x]<>-1
6310        THEN
6320          plot(x,wert[x])
6330        ELSE
6340          BEGIN
6350            plot(x,wert[x-1]);
6360            wert[x]:=wert[x-1];
6370          END; (* else *)
6380  END; (* andersverbinden *)
6390
6400
6410  PROCEDURE kopiewunsch;
6420
6430  { Hardcopy gewuenscht? }
6440
6450  BEGIN
6460    gotoxy(0,12);WRITE('Kopie ? (J,N)');
6470    IF jaantwort
6480      THEN
6490        copy
6500      ELSE
6510        delay(5000);
6520  END; (* kopiewunsch *)
6530
6540
6550  FUNCTION wiederholen:BOOLEAN;
6560
6570  { fragt, ob wiederholt werden soll }
6580
6590  BEGIN
6600    gotoxy(0,12);WRITE('wiederholen? (J/N)');
6610    wiederholen:=jaantwort;
6620  END; (* wiederholen *)
6630
6640
```

```
6650  BEGIN (* hauptprogramm *)
6660    REPEAT
6670      eingabemenue(umax,uachse,iachse,imax);
6680      werteeinlesen; kopiewunsch;
6690      berechnen;
6700      IF steil
6710        THEN
6720          BEGIN
6730            neuberechnen;
6740            andersverbinden;
6750          END (* if *)
6760        ELSE
6770          BEGIN
6780            lueckenweg;
6790            glaetten(0.49);
6800            glaetten(0.50);
6810            zeichnen;
6820          END; (* else *)
6830      verbinden; kopiewunsch;
6840    UNTIL NOT wiederholen;
6850    REPEAT
6860    UNTIL readkey<>CHR(0)                    (* bis eine Taste gedrueckt wird     *)
6870  END. (* kennlinie *)
```